PSA、ESA
Trace 取得 サンプルプログラム
説明書

2012 年 5 月作成　Ver.1.2

目次

1. 初めに

◆ このサンプルプログラムは、下記を使用して作成しています。

・Windows® Office 2007

・Microsoft® Excel® の Visual Basic® for Applications（VBA）

※Windows、Microsoft、Excel、Visual Basic は、米国およびその他の国における Microsoft Corporation の登録商標です。

◆ 簡単な設定、Trace データ取得のご参考資料としてご提供させていただくため、保証は行っておりません。ご了承ください。 実際のご利用にあたりましては、プログラムの内容を十分ご理解いただいた上で、お客様の責任で実行してください。

◆ このサンプルプログラムは、設定、トレースデータ取得の一例となっています。 最適のフローや TraceData の処理方法を示すものではございません。適宜ご使用目的に合わせた処理を行っていただくようお願いいたします。

◆ このサンプルプログラムは、スペクトラムアナライザモードの Swept SA 測定モードでご使用いただけるプログラムとなっています。 下記状態でご使用ください。

・[MODE]>{Spectrum Analyzer}を選択

・[MEASURE]>{Meas Off}

◆ 本プログラムは、Agilent IO Libraries Suite に含まれている VISA COM ライブラリを使用しています。 ご使用いただくには IO Libraries Suite を下記よりダウンロードしてインストールを行ってください。

http://www.agilent.com/find/iosuite

ページの、『IO ライブラリ XX.X を今すぐダウンロードしてください』というリンクよりダウンロードをして PC にインストールを行ってください。 （図 1.1）

ダウンロード、インストールの詳細につきましては、下記資料をご参照ください。

『Agilent IO Libraries Suite 16.2 簡易取扱説明書』

http://cp.literature.agilent.com/litweb/pdf/5991-0331JAJP.pdf

図 1.1

◆ VISA COM ライブラリを使用するには、下記方法で VISA COM 3.0 Type Library にチェックを入れてください。

1) XA VBA サンプルのエクセルファイルを開きます。

2) 開発タブを選択します。
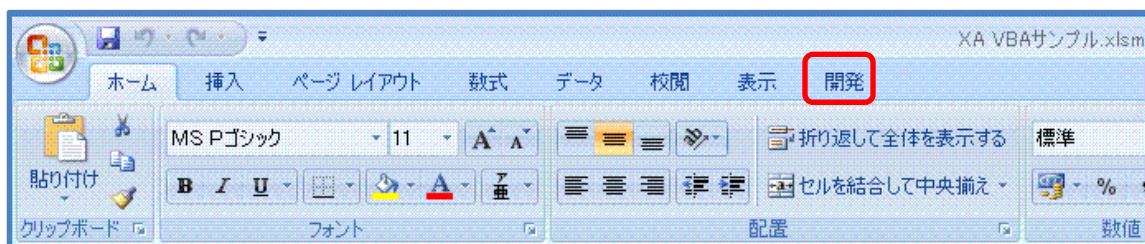
図 1.2

開発タブがない場合には、左上の Office ボタンから『**Excel のオプション**』を選択し、『**基本設定**』の中から『**[開発]タブをリボンに表示する**』にチェックを入れてください。 (図 1.3 図 1.4)
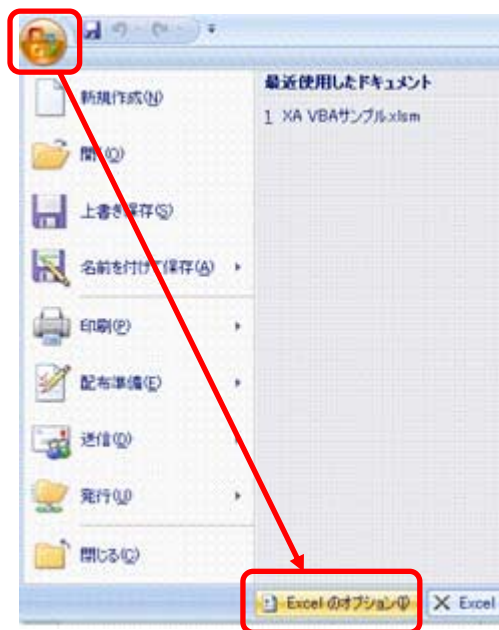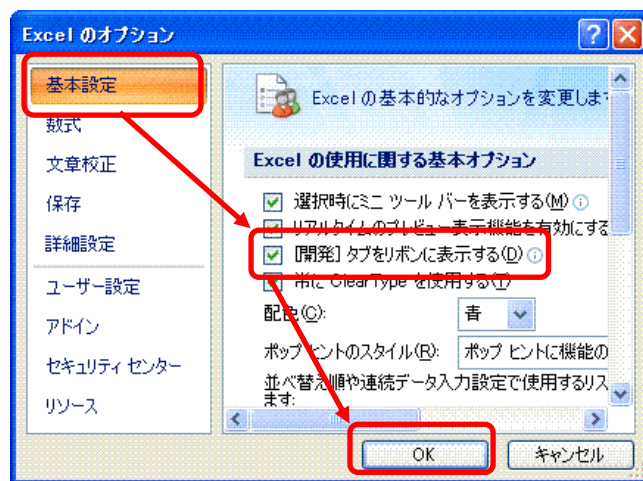
図 1.3　　　　　　　　　　　　図 1.4

3)　『Visual Basic』を押し、Microsoft Visual Basic の Window が開いたら、

　　ツール　→　参照設定

　を選択します。
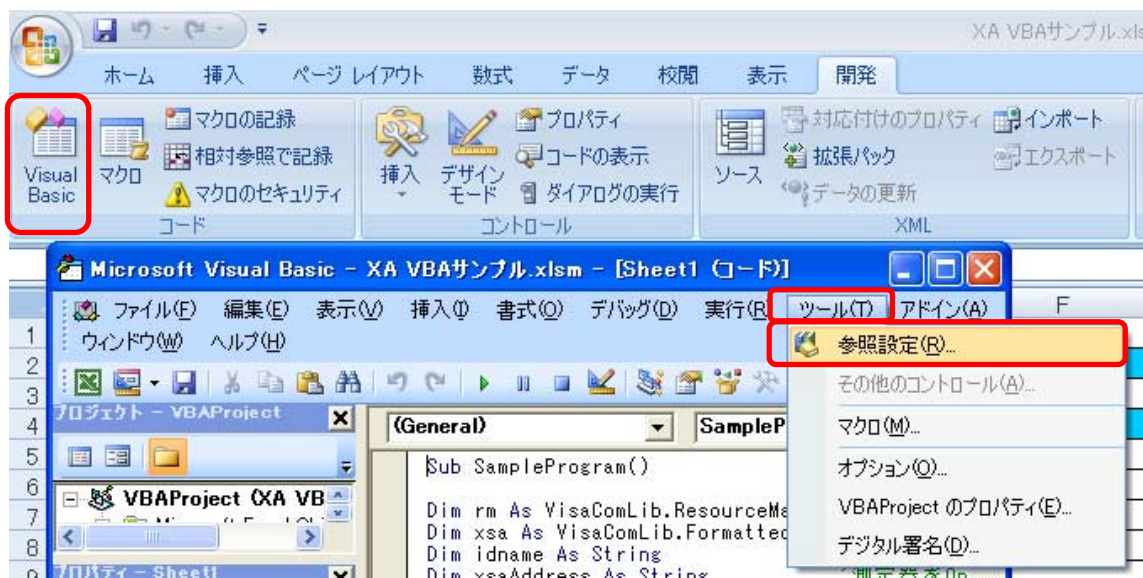
図 1.5

4) 『VISA COM 3.0 Type Library』にチェックを入れて『OK』を押してください。
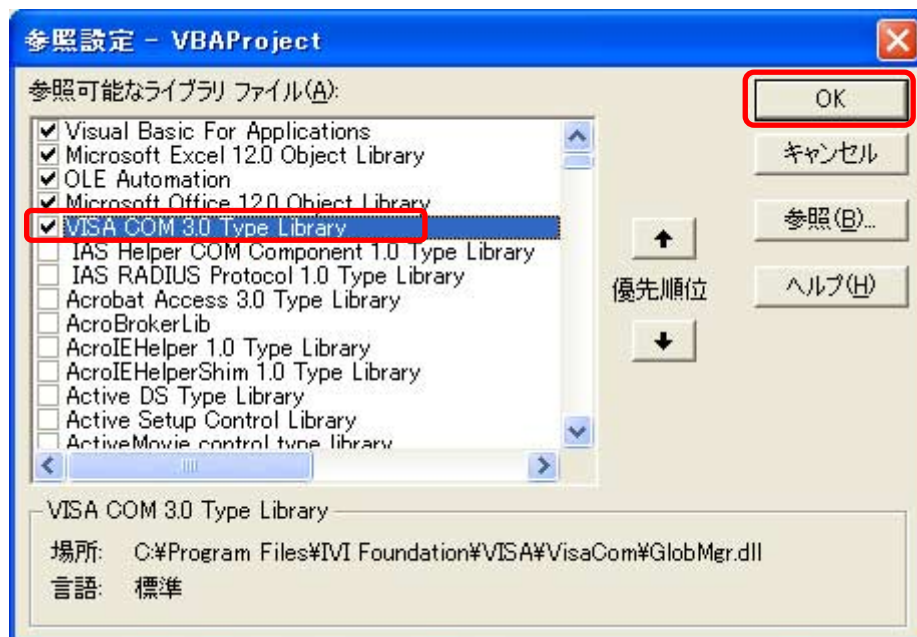
図 1.6

◆ VISA COM ライブラリの情報は、IO Libraries Suite のオンラインヘルプからご参照いただけます。 IO Libraries Suite をインストールすると、Windows のタスクバーの右端(画面の右下)に「IO」のアイコンが表示されます。

IO Libraries Version16.2 未満 ⇒

IO Libraries Version16.2 以上 ⇒

このアイコンをクリックしてメニューを表示させ、Documentation ＞＞ API Documentation ＞ VISA COM Help を実行してください。

2. プログラムの実行方法

1） PSAESA VBA サンプルのエクセルファイルを開きます。

2） Macro を ON にします。

セキュリティの警告　の部分に表示されている『Option』をクリックします。(図 2.1)

『セキュリティオプション』の Window が出たら、『このコンテンツを有効にする』を選択して『OK』を押します。(図 2.2)
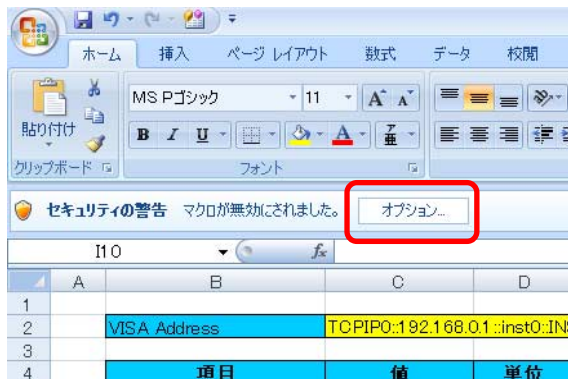
図 2.1
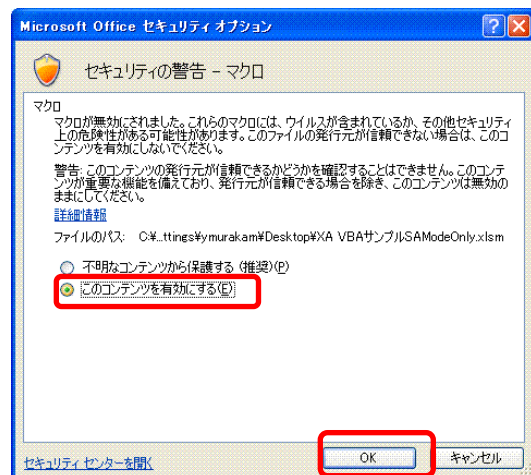
図 2.2

3)エクセルの黄色い部分が入力必須項目となります。

測定器の Address は VISA　Address を入力します。

接続方法によって下記のような形になります。

例)・LAN の場合　　TCPIP0::192.168.0.1::inst0::INSTR

・GPIB の場合　GPIB0::18::INSTR

・USB の場合　　USB0::0x0957::0x0A07::MYxxxxxxxx::0::INSTR　(IO Libraries　Version　16.2 以降)

USB0::2391::2567:: MYxxxxxxxx::0::INSTR　(IO Libraries　Version　16.2 未満)

VISA Address の確認方法については、付録 A を参照してください。

VISA Address	GPIB0::18::INSTR		接続確認

項目	値	単位	備考
Frequency	1G	Hz	G,M,kで記入
Span	1M	Hz	G,M,kで記入
Ref Level	-30	dBm	
Attenuation	20	dBm	
RBW	100k	Hz	M,kで記入
VBW	100k	Hz	M,kで記入
Detector	POSitive		AVERage,NEGative,POSitive,SAMPleより選択
Sweep Time Auto	OFF		ON OFFより選択
Sweep Time	10m	s	Sweep Time AutoがONの場合には無効
Sweep Points	101		
Trace	WRITe		WRITe,MAXHold,MINHoldより選択
Average Number	10		Average ON/OFFがOFFの場合は無効
Average ON/OFF	OFF		

図 2.3

4)黄色い部分の入力がすべて終わったら、『設定　&　トレースデータ取得』を押すと、設定と測定が実行されます。 測定を実行すると、掃引モードは自動で Single 掃引となり、掃引を1回した状態でトレースデータを取得して止まります。

実行後、は下記のようなデータが表示されます。

・『接続確認』の部分に、測定器の型番、シリアル番号、Firmware 番号が表示されます。

・『エラー確認』の部分にエラーの有無が表示されます

・『Trace Data』の部分に、トレースデータ(レベルのみ)が表示されます。

VISA Address	GPIB0::18::INSTR		接続確認	Hewlett-Packard, E4407B, US012345678, A14.06

項目	値	単位	備考
Frequency	1G	Hz	G,M,kで記入
Span	1M	Hz	G,M,kで記入
Ref Level	-30	dBm	
Attenuation	20	dBm	
RBW	100k	Hz	M,kで記入
VBW	100k	Hz	M,kで記入
Detector	POSitive		AVERage,NEGative,POSitive,SAMPleより選択
Sweep Time Auto	OFF		ON OFFより選択
Sweep Time	10m	s	Sweep Time AutoがONの場合には無効
Sweep Points	101		
Trace	WRITe		WRITe,MAXHold,MINHoldより選択
Average Number	10		Average ON/OFFがOFFの場合は無効
Average ON/OFF	OFF		

図 2.4

3. 使用しているコード、コマンドの説明

本プログラムのコードは下記のようになっています。

++

```
Sub SampleProgram()

Dim rm As VisaComLib.ResourceManager
Dim sa As VisaComLib.FormattedIO488
Dim idname As String                 '*IDN?の戻り値を格納
Dim saAddress As String              '測定器を Open にするときの Address をエクセルから取得
Dim opcvalue As String               '*OPC? の戻り値を格納
Dim error As String                  'SYST:ERR?の戻り値を格納
Dim dbSwepoin As Double              'スイープポイント数取得時の戻り値を格納
Dim dbTraceData() As Double          '取得したトレースデータを格納
Dim MaxRowE As String                'セル初期化に使用　E 列の古いデータの最後のセルを格納

Set rm = New VisaComLib.ResourceManager
Set sa = New VisaComLib.FormattedIO488

saAddress = Range("C2")

Set sa.IO = rm.Open(saAddress)

'Timeout を設定
sa.IO.Timeout = 10000

'*IDN?クエリを送り、測定機との接続を確認します。
sa.WriteString "*IDN?"
idname = sa.ReadString()
Cells(2, 7) = idname

'*CLS 実行  StatusBite、 エラーのクリア
sa.WriteString "*CLS"

'Preset 実行
sa.WriteString ":SYST:PRES"
```

```
'*OPC?にて Preset 完了待ち
sa.WriteString "*OPC?"
opcvalue = sa.ReadString

'Sweep Single 設定
sa.WriteString ":INIT:CONT OFF"

'Frequency 設定
sa.WriteString ":FREQ:CENT " & Worksheets("Sheet1").Range("C5").value & "Hz"

'Span 設定
sa.WriteString ":FREQ:SPAN " & Worksheets("Sheet1").Range("C6").value & "Hz"

'Ref Level 設定
sa.WriteString ":DISP:WIND:TRAC:Y:RLEV " & Worksheets("Sheet1").Range("C7").value &
"dBm"

'Attenuation 設定
sa.WriteString ":POW:ATT " & Worksheets("Sheet1").Range("C8").value

'RBW 設定
sa.WriteString ":BAND " & Worksheets("Sheet1").Range("C9").value & "Hz"

'VBW 設定
sa.WriteString ":BAND:VID " & Worksheets("Sheet1").Range("C10").value & "Hz"

'Detector 設定
sa.WriteString ":DET " & Worksheets("Sheet1").Range("C11").value

'Sweep Time 設定
sa.WriteString ":SWE:TIME " & Worksheets("Sheet1").Range("C13").value & "s"

'Sweep Time Auto 設定
sa.WriteString ":SWE:TIME:AUTO " & Worksheets("Sheet1").Range("C12").value
```

```
'Sweep Points 設定
sa.WriteString ":SWE:POIN " & Worksheets("Sheet1").Range("C14").value

'Trace 設定
sa.WriteString ":TRAC:MODE " & Worksheets("Sheet1").Range("C15").value

'Average Number 設定
sa.WriteString ":AVER:COUN " & Worksheets("Sheet1").Range("C16").value

'Average/Hold Number 設定
sa.WriteString ":AVER " & Worksheets("Sheet1").Range("C17").value


'測定トリガ
sa.WriteString ":INIT:IMM"

'*OPC?にて設定完了待ち
sa.WriteString "*OPC?"
opcvalue = sa.ReadString

'エラー確認
sa.WriteString ":SYST:ERR?"
error = sa.ReadString
Cells(24, 2) = error


''''''ここから TraceData 取得''''''

'ポイント数取得 データがいくつ返ってくるかを把握するために使用
sa.WriteString ":SWE:POIN?"
dbSwepoin = sa.ReadNumber()

'トレースデータを格納する dbTraceData の配列に含まれる変数の数の変更
ReDim dbTraceData(dbSwepoin-1)

'トレースデータ取得
```

```
sa.WriteString ":TRAC:DATA? TRACE1"
dbTraceData = sa.ReadList(ASCIIType_R8, ",")


'トレースデータ表示部分　過去のデータを削除
MaxRowE = Range("E24").End(xlDown).Row
Range("E24:E" & MaxRowE).Clear


'トレースデータ表示処理
For i = 0 To dbSwepoin - 1
  Range("E" & 24 + i).value = CDbl(dbTraceData(i))
Next i


sa.IO.Close
Set sa = Nothing
Set rm = Nothing


End Sub
```

++

以下、各コマンドの詳細を記載いたします。

◆変数の宣言と測定器の接続◆

```
Sub SampleProgram()

Dim rm As VisaComLib.ResourceManager
Dim sa As VisaComLib.FormattedIO488
Dim idname As String              '*IDN?の戻り値を格納
Dim saAddress As String           '測定器をOpenにするときのAddressをエクセルから取得
Dim opcvalue As String            '*OPC? の戻り値を格納
Dim error As String               'SYST:ERR?の戻り値を格納
Dim dbSwepoin As Double           'スイープポイント数取得時の戻り値を格納
Dim dbTraceData() As Double       '取得したトレースデータを格納
Dim MaxRowE As String             'セル初期化に使用　E列の古いデータの最後のセルを格納

Set rm = New VisaComLib.ResourceManager
Set sa = New VisaComLib.FormattedIO488

saAddress = Range("C2")

Set sa.IO = rm.Open(saAddress)

'Timeoutを設定
sa.IO.Timeout = 10000
```

上記の赤枠で囲っているところは VISA COM を参照し、Address を指定して測定器とのセッションを開始するのに関係する部分となります。

saAddress = Range(“C2”)

セルの C2 から測定器の VISA　Address を読み込み saAddress に格納します。

sa.IO.Timeout = 10000

TimeOutを設定しています。

その他、宣言している変数の詳細についてはプログラムのコメントを参照してください。

◆初期設定、接続先確認◆

```
'*IDN?クエリを送り、測定機との接続を確認します。
sa.WriteString "*IDN?"
idname = sa.ReadString()
Cells(2, 7) = idname

'*CLS実行 StatusBite、 エラーのクリア
sa.WriteString "*CLS"

'Preset実行
sa.WriteString ":SYST:PRES"

'*OPC?にてPreset完了待ち
sa.WriteString "*OPC?"
opcvalue = sa.ReadString

'Sweep Single設定
sa.WriteString ":INIT:CONT OFF"
```

sa.WriteString "*IDN?"

＊IDN?というコマンドを測定器に送ります。 PSA/ESAスペクトラムアナライザでは＊IDN?クエリを使用すると測定器のモデル名、シリアル番号、FirmwareVersionが確認できます。

idname = sa.ReadString()

＊IDN?の戻り値を読み込んで idname に格納します。

Cells(2, 7) = idname

Idnameに格納した*IDN?の戻り値を、セルの2行7列目に表示します。

sa.WriteString "*CLS"

＊CLSというコマンドを測定器に送ります。 ＊CLSでは、今まで測定器内に残っていたエラーキューのクリアと、Status Byte Registerのクリアを行います。

sa.WriteString "SYST:PRES"

SYST:PRESいうコマンドを測定器に送ります。 SYST：PRESコマンドは、測定器のプリセットを行います。 測定器本体の[reset]というボタンを押した状態になります。

sa.WriteString "*OPC?"

＊OPC?というコマンドを測定器に送ります。 ＊OPC?は前に送ったコマンド動作が完了すると、1を測定器が返すコマンドになります。 Presetや掃引トリガ後など、処理に時間がかかるコマンドを送った後などには＊OPC?で動作完了を待ってから次のフローへ進むようにしていただくことをお勧めします。

opcvalue = sa.ReadString

Preset動作が完了した後、測定器が返してきた1を受け取り、opcvalueへ格納します。

sa.WriteString "INIT:CONT OFF"

INIT:CONT OFFいうコマンドを測定器に送ります。 INIT:CONT OFFは測定をSingle測定に切り替えるコマンドになります。 測定器の[Single]ボタンを押したときと同じ動作になります。 逆に、測定をContinuousにしたい場合には、 INIT:CONT ON というコマンドを使用します。

◆各パラメタの設定◆

```
'Frequency設定
sa.WriteString ":FREQ:CENT " & Worksheets("Sheet1").Range("C5").value & "Hz"

'Span設定
sa.WriteString ":FREQ:SPAN " & Worksheets("Sheet1").Range("C6").value & "Hz"

'Ref Level設定
sa.WriteString ":DISP:WIND:TRAC:Y:RLEV " & Worksheets("Sheet1").Range("C7").value & "dBm"

'Attenuation設定
sa.WriteString ":POW:ATT " & Worksheets("Sheet1").Range("C8").value

'RBW設定
sa.WriteString ":BAND " & Worksheets("Sheet1").Range("C9").value & "Hz"

'VBW設定
sa.WriteString ":BAND:VID " & Worksheets("Sheet1").Range("C10").value & "Hz"

'Detector設定
sa.WriteString ":DET " & Worksheets("Sheet1").Range("C11").value

'Sweep Time設定
sa.WriteString ":SWE:TIME " & Worksheets("Sheet1").Range("C13").value & "s"

'Sweep Time Auto設定
sa.WriteString ":SWE:TIME:AUTO " & Worksheets("Sheet1").Range("C12").value

'Sweep Points設定
sa.WriteString ":SWE:POIN " & Worksheets("Sheet1").Range("C14").value

'Trace設定
sa.WriteString ":TRAC:MODE " & Worksheets("Sheet1").Range("C15").value

'Average Number設定
sa.WriteString ":AVER:COUN " & Worksheets("Sheet1").Range("C16").value

'Average/Hold Number設定
sa.WriteString ":AVER " & Worksheets("Sheet1").Range("C17").value


'測定トリガ
sa.WriteString ":INIT:IMM"

'*OPC?にて設定完了待ち
sa.WriteString "*OPC?"
opcvalue = sa.ReadString
```

エクセルの黄色い部分で指定した設定を測定器に設定し、設定コマンドをすべて送ったところで測定トリガをかけて測定を開始し、＊OPC?で掃引完了を待ちます。

各項目の設定コマンドについては下記のようになっています。

++周波数設定++

※PSA/ESA共通

　コマンド　[:SENSe]:FREQuency:CENTer <freq>

プログラム中での表記では、下記のようになっています。

　sa.WriteString ":FREQ:CENT " & Worksheets("Sheet1").Range("C5").value & "Hz"

sa.WriteString　で以下に続く部分を測定器にコマンドとして送ります。

まず、""でかこまれた　FREQ:CNET[space]　文字列があり、そのあと&で続くWorksheets("Sheet1").Range("C5").valueの内容が組み込まれ、最後にまた&で""でくくられたHz

が文字列としてついたものが一つのコマンドとして送られます。
Worksheets("Sheet1").Range("C5").value は、エクセルのシート1のC5のセルの内容を読み込むことになります。 この部分についてはVBAの言語の内容になりますのでVBA言語の資料などご参照ください。

コマンドの、大文字、小文字、[ ]などの部分については『付録Cコマンドの見方と書き方』を御参照ください。

たとえば、エクセルのFrequencyの行の黄色い部分に 1G と入力した場合、実際に送っているコマンドは、 :FREQ:CENT 1GHz となります。

以下、細かい書き方は同じになりますので、各コマンドについて記載いたします。

++Span設定++
※PSA/ESA共通
コマンド [:SENSe]:FREQuency:SPAN <freq>

++Ref Level設定++
※PSA/ESA共通
コマンド :DISPlay:WINDow:TRACe:Y:[SCALe]:RLEVel <ampl>

++Attenuation設定++
※PSA/ESA共通
コマンド [:SENSe]:POWer[:RF]:ATTenuation <rel_ampl>

++RBW設定++
※PSA/ESA共通
コマンド [:SENSe]:BANDwidth|BWIDth[:RESolution] <freq>

++VBW設定++
※PSA/ESA共通
コマンド [:SENSe]:BANDwidth|BWIDth:VIDeo <freq>

++Detector設定++

※PSA

コマンド [:SENSe]:DETector[:FUNCtion]

NORMal|AVERage|POSitive|SAMPle|NEGative|QPEak|EAVerage|EPOSitive|MPOSitiv|RMS

※ESA

コマンド [:SENSe]:DETector[:FUNCtion] NEGative|POSitive|SAMPle|AVERage|RMS

プログラム中では、AVERage,NEGative,POSitive,SAMPleしか使用しておりませんが、使用する測定器によって下記Detectorがご使用いただけます。

PSA		ESA(OptAYQなし)	
Detector	コマンド書き方	Detector	コマンド書き方
Normal	NORMal	Average	AVERage
Average	AVERage	Peak	POSitive
Peak	POSitive	Negative peak	NEGative
Sample	SAMPle	Sample	SAMPle
Negative peak	NEGative	RMS(Average TypeがRMS)	RMS
Quasi Peak	QPEak		
EMI Average	EAVerage		
EMI Peak	EPOSitive		
MIL Peak	MPOSitive		
RMS Average (Average TypeがRMS)	RMS		

※ESA OptionAYQつきの場合

コマンド [:SENSe]:DETector[:FUNCtion]:EMI QPEak|AVERage|OFF

++Sweep Time設定++

※PSA/ESA共通

コマンド [:SENSe]:SWEep:TIME <time>

++Sweep Time Auto設定++

※PSA/ESA共通

コマンド [:SENSe]:SWEep:TIME:AUTO OFF|ON|0|1

Sweep TimeをAutoにした後にSweepTimeを設定してしまうと、自動でSweep Time AutoがOFFとな

ってしまうため、プログラムでは、Auto設定をTimeの設定より後にするようにしています。
プログラムのフローで、SweepTimeAuto がONの場合には、SweepTimeの設定コマンド自体を送らないようにしていただくのもよいかと思います。

++Sweep Points設定++
※PSA/ESA共通
　コマンド　[:SENSe]:SWEep:POINts <integer>

++Trace設定++
※PSA/ESA共通
　コマンド　:TRACe1|2|3:MODE WRITe|MAXHold|MINHold|VIEW|BLANk
本プログラムでは、Trace1のみ制御しているため、TRAC:MODE・・・となっています。

++Average/Hold Number設定++
※PSA/ESA共通
　コマンド　[:SENSe]:AVERage:COUNt <integer>

++Average　ON/OFF設定++
※PSA/ESA共通
　コマンド　[:SENSe]:AVERage[:STATe] OFF|ON|0|1

◆エラー確認◆

```
'エラー確認
sa.WriteString ":SYST:ERR?"
error = sa.ReadString
Cells(24, 2) = error
```

sa.WriteString ":SYST:ERR?"
　設定を送り、掃引をしたところでエラー確認をしています。エラーを確認するコマンドは、下記になります。
　:SYSTem:ERRor[:NEXT]?

このコマンドは、出ているエラーを返すものですが、複数エラーが出た場合でも、一番履歴の古いエラー1つしか返しません。 エラーを返すと、返したエラーは内部的にクリアされ、次にもう一度SYST:ERR?を送ると次に古いエラーを1つだけ返します。

たとえば　最初に

　-222,″Data out of range;value clipped to upper limit.″

というエラーが出ていて、次に、

　-410,″Query INTERRUPTED″

というエラーが出ていた場合、

:SYST:ERR？を一度送ると“-222,″Data out of range;value clipped to upper limit.″”が返ってきて、もう一度:SYST:ERR？を送ると、“-410,″Query INTERRUPTED″”が返ってくるような動作となります。

　何もエラーがないときは、 “+0,″No error″”が返ってくるので、通常は、 +0,″No error″が返ってきたら問題ないと判断するプログラムフローなどにしていただくなどが良いかと思います。

error = sa.ReadString

　:SYST:ERR?の戻り値を読み込み、errorに格納しています。

Cells(24, 2) = error

　この部分で、Cellの24列2行にSYST:ERR?の戻り値を表示させています。

◆トレースデータ取得◆

```
'''''''ここからTraceData取得''''''''

'ポイント数取得 データがいくつ返ってくるかを把握するために使用
sa.WriteString ":SWE:POIN?"
dbSwepoin = sa.ReadNumber()

'トレースデータを格納するvTraceDataの配列に含まれる変数の数の変更
ReDim dbTraceData(dbSwepoin - 1)

'トレースデータ取得
sa.WriteString ":TRAC:DATA? TRACE1"
dbTraceData = sa.ReadList(ASCIIType_R8, ",")
```

sa.WriteString ″:SWE:POIN?″

　表示ポイント数を取得しています。ポイント数を確認するコマンドは、下記になります。

　[:SENSe]:SWEep:POINts?

dbSwepoin = sa.ReadNumber()

　:SWE:POIN?の戻り値を読み込み、dbSwepoinに格納しています。

この値は、後でTrace Dataを加工する際などに使用しています。

ReadNumber()メソッドを使用すると、「一つの数値データ」はDouble として読み込めます。

ReDim dbTraceData (dbSwepoin-1)

　dbTraceDataの配列に含まれる変数の数を変更しています。 ここで、:SWE:POIN?で取得したポイント数を使用しています。

sa.WriteString ":TRAC:DATA? TRACE1"

　Trace Dataを取得します。 Trace Dataを取得するコマンドは下記になります。

　:TRACe[:DATA]? TRACE1 | TRACE2 | TRACE3

このコマンドを使用すると、Trace Dataのレベルのみが取得できます。 戻り値は、1ポイント目のレベル、2ポイント目のレベル、3ポイント目のレベル・・・ といった順序になっており、データはカンマ区切りになっています。

ESA、PSAスペクトラムアナライザでは、TraceDataとして取得できるのはレベルデータのみとなっています。 周波数データはコマンドでは取得できないため、周波数のStart/Stopとポイント数からプログラム側で計算していただくようなフローとしていただく必要があります。

周波数のStart/Stopは下記コマンドで取得できます。

++Start周波数++

　[:SENSe]:FREQuency:STARt?

++Stop周波数++

　[:SENSe]:FREQuency:STOP ?

dbTraceData = sa.ReadList(ASCIIType_R8, ",")

　:TRAC:DATA? TRACE1の戻り値をdbTraceDataに格納しています。 ReadList()メソッドを使用すると「カンマ区切りの複数の数値データ」は、指定したセパレータ（デフォルトはカンマ）で指定した型の配列として読み込めます。

◆Trace Data表示部分のデータの削除◆

```
'トレースデータ表示部分　過去のデータを削除
MaxRowE = Range("E24").End(xlDown).Row
Range("E24:E" & MaxRowE).Clear
```

dbTraceDataのデータをエクセルに表示させるのですが、前に実行したときの結果が残ってしまっているため、一度セルをきれいにしています。　データがある最後の行を探して、E列の24行目からデータのある最後の行までをクリアしています。

Range().End(xlDown).Rowや、Range().Clearのご使用方法につきましてはVBAのコマンドになりますので、VBA言語の資料などご参照ください。

◆Trace Data表示処理◆

```
'トレースデータ表示処理
For i = 0 To dbSwepoin - 1
 Range("E" & 24 + i).value = CDbl(dbTraceData(i))
Next i
```

dbTraceDataの値をエクセルのE列に表示させています。

◆クローズ処理とEnd Sub◆

```
sa.IO.Close
Set sa = Nothing
Set rm = Nothing
```

クローズ処理をしています。

付録A VISA Address の確認方法

まず IO Libraries の Connection Expert を立ち上げます。タスクバーにある IO Libraries のアイコン

（ (Version16.2 未満)または (Version16.2 以上)）をクリックし、ポップアップされるメニューから

『Agilent Connection Expert』を選択します。 接続方法によってそれぞれ下記のように確認します。

◆LANで接続している場合

1）ツリーから『LAN（TCPIP0）』をクリックして選択し、『Add Instrument』を押します。
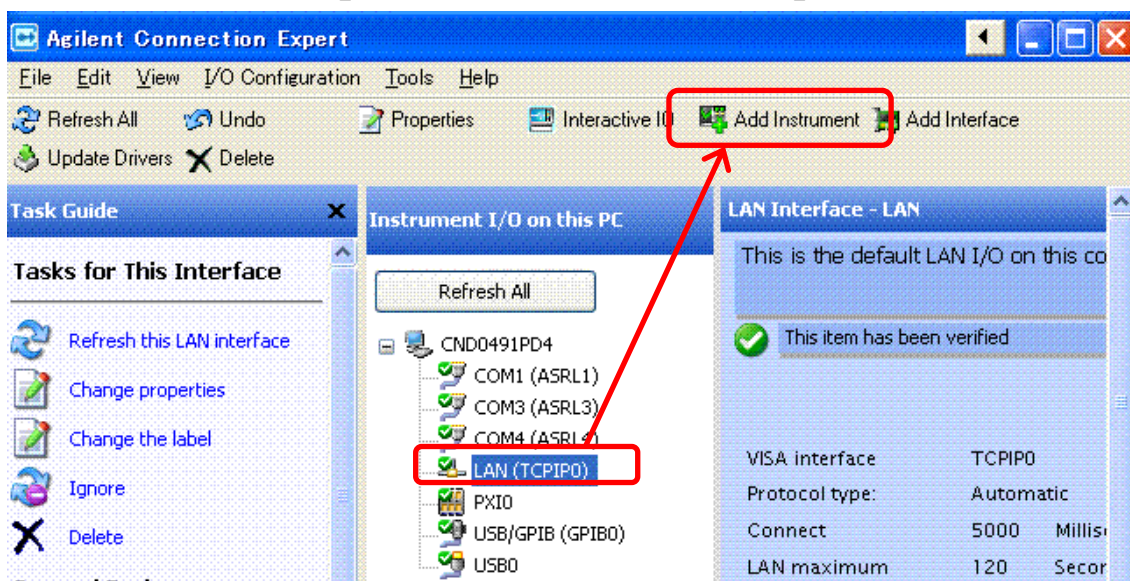
図 A.1

2)『Add Address』を押し、『Use IP Address』にチェックを入れて、測定器の Address を入力します。 (図 A.2 ①、②参照)

Instrument Identification の「Web information(recommended)を「＊IDN query」に変更して、『Test Connection』を押し（図 A.2 ③、④参照）、右側に緑のチェックマークがついたら『OK』を押します。 (図 A.2 ⑤参照)

図 A.2

3) ツリーに測定器が追加されます。 追加された測定器をクリックした状態で右の画面に出てくる VISA Address: が測定器の VISA Address となります。

図 A.3

◆GPIB（USB/GPIB を含む）、USB で接続している場合

Agilent Connection Expert の真ん中に表示されているツリーに、認識されている測定器が表示されます。　測定器が表示されていない場合には、『Refresh All』を押してみてください。

ツリーに表示されている測定器をクリックした状態で右の画面に出てくる　VISA Address:　が測定器の VISA Address となります。
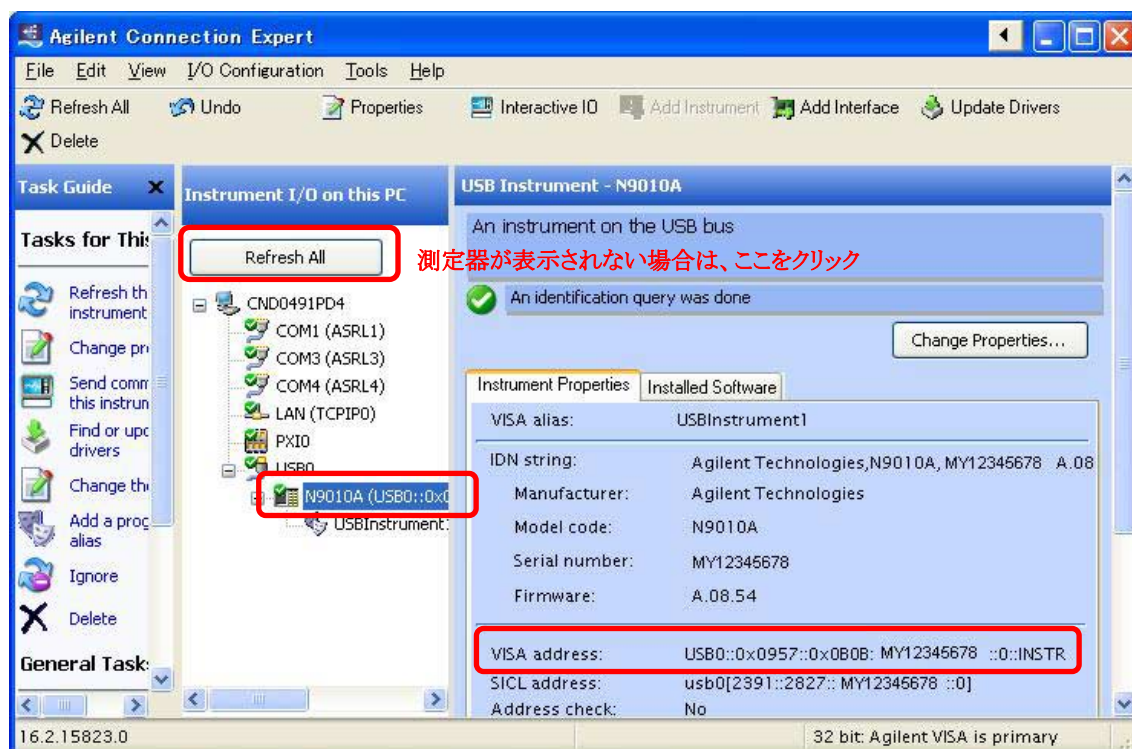
図 A.4

付録 B　Help を使用したコマンドの参照方法

PSA 、ESA スペクトラムアナライザでは、本体の画面左下にある、[Help]を使用すると、操作に該当するコマンドが確認いただけるようになっています。

図 B.1 [Help]を押した状態
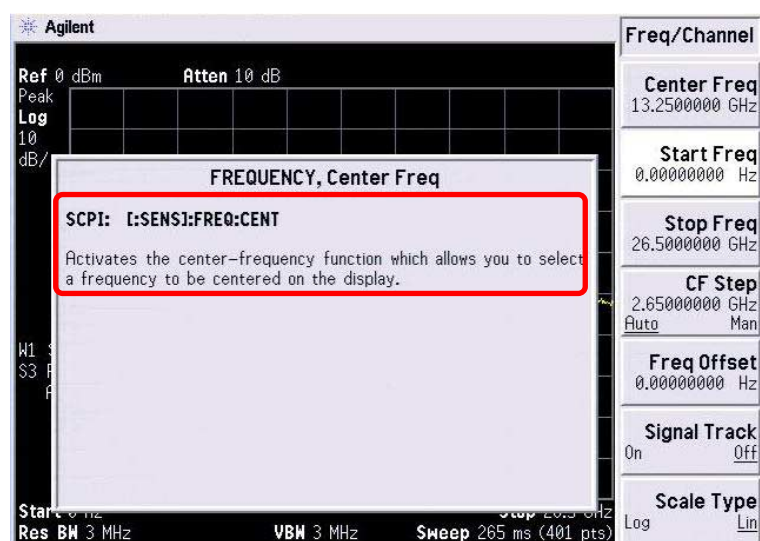
図 B.2 [Help]の後に[Center Freq]を押した状態

該当操作の Help 画面が開いた状態（図 B.2）で別のボタンを押しますと Help 機能が解除され、通常操作に戻ります。

付録C　コマンドの見方と書き方

　Help やマニュアルをご覧いただくと、コマンドが、下記のように小文字や大文字、|、[、]などが使われて書かれています。

　[:SENSe]:DETector:TRACe[1]|2|3|4|5|6
AVERage|NEGative|NORMal|POSitive|SAMPle|QPEak|EAVerage|RAVerage

まず、**大文字と小文字**ですが、小文字は省略可能な文字となっています。たとえば下記の場合、

　:FREQuency:CENTer

小文字は省いても良いので、下記のように書けることになります

　:FREQ:CENT

次に、**[ ]で囲まれた部分**は省略可能な部分となります。たとえば下記の場合

　[:SENSe]:FREQuency:SPAN

[ ]と小文字を省くことができるので、下記のように書けることになります。

　:FREQ:SPAN

ただし、[ ]の部分を省略せずに書く場合、[ ]はコマンドとしては書きませんので下記のようになります。

　:SENS:FREQ:SPAN

|で仕切られているものは、いずれかを選択することを示します。たとえば下記の場合、

　[:SENSe]:DETector:TRACe[1]|2|3|4|5|6
AVERage|NEGative|NORMal|POSitive|SAMPle|QPEak|EAVerage|RAVerage

まず、Traceの1から6のいずれかを選びます。 また、DetectorのTypeをAverage、Negative・・・の中から一つ選びます。 たとえばTrace2に関する処理で、DetectorをNegativeにしたい場合は、

　:DET:TRAC2 NEG

となります。 さらにTrace1の場合は[ ]となっていて省略可能なため、Trace1に関する処理に限り下記のようにかけることになります。

　:DET:TRAC NEG

付録 D　よく使用されるコマンド

◆[MEASURE]モードの切り替え

　CONF:<測定モード>

または、

　INITiate: <測定モード>

CONF コマンドを使用すると Measure モードを切り替えると同時に Preset が行われ、すべての設定が Default 状態となります。 <測定モード>の部分にはそれぞれ下記が入ります。

※CONF:SAN のみは、Default 状態に戻りません。 Meas Off 状態に CONF:SAN で戻した後、SYST:PRES コマンドをご使用いただきますと、プリセット状態に戻ります。

Measure	<測定モード>の記述	Measure	<測定モード>の記述
Meas Off	SANalyzer	Harmonic Distortion	HARMonics
Channel Power	CHPower	Burst Power	BPOWer
Occupied BW	OBWidth	Intermod (TOI)	TOI
ACP	ACPower	Spurious Emissions	SPURious
Power Stat CCDF	PStatistic	Spectrum Emission Mask	SEMask
Multi Carrier Power ※PSA のみ	MCPower		

たとえば、Channel Power に切り替えたいときは、下記のようになります。

　CONF:CHP

◆Trace Data を測定器内部に保存するコマンド

　下記コマンドを使用すると、Trace Data を測定器内部に保存することができます。

　　:MMEMory:STORe:TRACe　TRACE1|TRACE2|TRACE3|ALL,<filename>

Trace の Format を『Trace+State』で保存するか『CSV』で保存するかは、<filename>の拡張子で変わります。 Trace+State はスペクトラムアナライザで読み込むための Trace となりますので PC などで開いてデータを見ることはできません。 逆に、CSV のデータは PC で Trace Data を見ることができますが、スペクトラムアナライザで読み込むことはできません。 拡張子は下記のように指定してください。

Format	拡張子
Trace+State	.trc
CSV	.csv

また、<filename>にはドライブ名を含めて記載します。 C ドライブはスペクトラムアナライザの内部メモリ、A ドライブはフロッピーディスクとなります。

たとえば、Trace2 を MYTRACE という File 名で A ドライブ（フロッピーディスク）に CSV ファイルで保存したい場合は下記のようになります。

:MMEM:STOR:TRAC TRACE2,″A:MYTRACE.csv″

File Name は文字列として送ってください。

◆画像データを測定器内部に保存するコマンド

下記コマンドを使用すると、画像を測定器内部に保存することができます。

:MMEMory:STORe:SCReen <filename>

<filename>にはドライブ名を含めて記載します。 C ドライブはスペクトラムアナライザの内部メモリ、A ドライブはフロッピーディスクとなります。

たとえば、 MYSCREEN というファイル名でフロッピーディスクに保存したい場合、下記のようになります。

:MMEM:STOR:SCR ″A:MYSCREEN.gif″

File Name は文字列として送ってください。

◆スペアナ内部に保存した Trace Data や画像データを PC に転送するコマンド

下記コマンドを使用すると、測定器内部に保存されている画像データや Trace Data などを PC に転送することができます。

:MMEMory:DATA? <file_name>

<filename>にはドライブ名を含めて記載します。 C ドライブはスペクトラムアナライザの内部メモリ、A ドライブはフロッピーディスクとなります。

たとえば、 MYSCREEN.gif というファイル名でフロッピーディスクに保存した画像を PC に転送したい場合は下記のようになります。

:MMEM:DATA? ″A:MYSCREEN.gif″

返ってくるデータは binblock フォーマットとなっています。

◆スペアナ内部に保存した Trace Data や画像データを削除するコマンド

　下記コマンドを使用すると、測定器内部に保存されている画像データや Trace Data などを削除することができます。

:MMEMory:DELete <file_name>

<filename>にはドライブ名を含めて記載します。 C ドライブはスペクトラムアナライザの内部メモリ、A ドライブはフロッピーディスクとなります。

たとえば、 MYTRACE.csv というファイル名でフロッピーディスクに保存されている File を削除したい場合は、下記のようになります。

:MMEM:DEL "A:MYTRACE.csv"

File Name は文字列として送ってください。

◆マーカーの値を取得するコマンド

　マーカーの周波数データ（X 軸）の値を取得するコマンドは下記になります。

:CALCulate:MARKer[1]|2|3|4:X?

マーカーのレベルデータ（Y 軸）のデータを取得するコマンドは下記になります。

:CALCulate:MARKer[1]|2|3|4:Y?

◆マーカーを Peak に移動するコマンド

Peak Search をするコマンドは下記になります。

:CALCulate:MARKer[1]|2|3|4:MAXimum

◆Marker を表示させるコマンド

Marker を表示させるコマンドは下記になります。

:CALCulate:MARKer[1]|2|3|4:MODE POSition|DELTa|BAND|SPAN

たとえば、Normal Marker1 を表示させたい場合は下記のようになります。

:CALC:MARK:MODE POS

◆任意の周波数にマーカーを置くコマンド

任意の周波数にマーカーを置くコマンドは下記になります。

:CALCulate:MARKer[1]|2|3|4:X <param>

たとえば、1GHzに Marker1 を移動させたい場合は、下記のようになります。

:CALC:MARK:X 1GHZ

Published in Japan, May 23,2012
5991-0298JAJP
0000-08A